Final_Proj

Sibo Zhu

2017/12/15

```
# set up connection
requestURL <- "https://api.twitter.com/oauth/request_token"
accessURL <- "https://api.twitter.com/oauth/access_token"
authURL <- "https://api.twitter.com/oauth/authorize"
api_key             <- "dSzLRlqni1W2f9CJagS19RP6b"
api_secret          <- "pLX7gIERogxsgn7AT2eEAQlhmxGKwEx9de5pjeOtjkQDDbVVCD"
access_token        <- "3134023545-OZ9TIdXFbEFW66HmsCUo0EWGIANJ4SgS1OgdDO7"
access_token_secret <- "kAzcVK0EMOhCFAs13Mau2Y8q2fCYHjvA5XXl7Rd0SCO1o"
######### Prepare for streamR
#my_oauth <- OAuthFactory$new(consumerKey = api_key, consumerSecret = api_secret,
#requestURL = requestURL, accessURL = accessURL, authURL = authURL)
#my_oauth$handshake(cainfo = system.file("CurlSSL", "cacert.pem", package = "RCurl"))
#save(my_oauth, file = "my_oauth.Rdata")
#load("my_oauth.Rdata")

######### Prepare for twitteR
#setup_twitter_oauth(api_key, api_secret, access_token, access_token_secret)
```

```
#load("my_oauth.Rdata")

## capture tweets mentioning the "realDonaldTrump" hashtag or sent from United States
#  filterStream( file="D_Tweet.json", track="realDonaldTrump",
#      locations=c(-74,40,-73,41), timeout=7200, oauth=my_oauth )
```

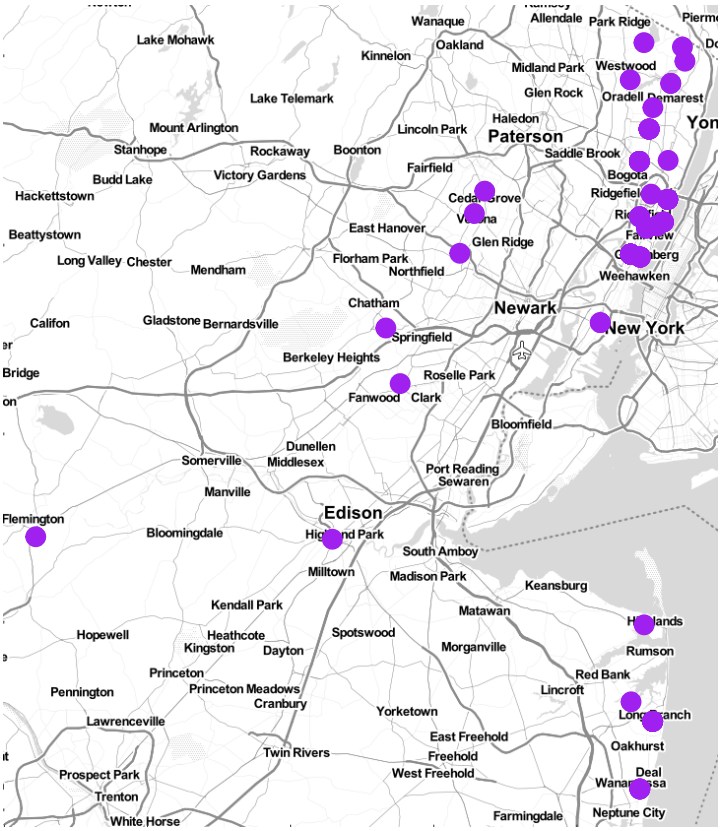
```
tweets_raw.df <- parseTweets("DT_Tweet.json",verbose = FALSE)


keep <- c("text","lang","listed_count","geo_enabled","statuses_count","followers_count",
          "favourites_count","friends_count","time_zone","country_code","full_name",
          "place_lat","place_lon")

tweets.df <- tweets_raw.df[,keep];


write.csv(tweets.df,"tweets_df.csv")
```

```
tweets.df <- tweets.df[tweets.df$lang=="en",]
tweets.df <- tweets.df[tweets.df$country_code=="US",]
tweets.df <- tweets.df[tweets.df$geo_enabled==TRUE,]
tweets.df <- tweets.df[tweets.df$place_lat >25 & tweets.df$place_lat <50 &
                       tweets.df$place_lon > -125 & tweets.df$place_lon< -66,]

write.csv(tweets.df,"cleaned_tweets_df.csv")


#filter with place
ca <- data.frame(filter(tweets.df, grepl(', CA', full_name)))
nj <- data.frame(filter(tweets.df, grepl(', NJ', full_name)))
```

```
ny <- data.frame(filter(tweets.df, grepl(', NY', full_name)))
fl <- data.frame(filter(tweets.df, grepl(', FL', full_name)))

#write.csv(il,"il.csv")
```

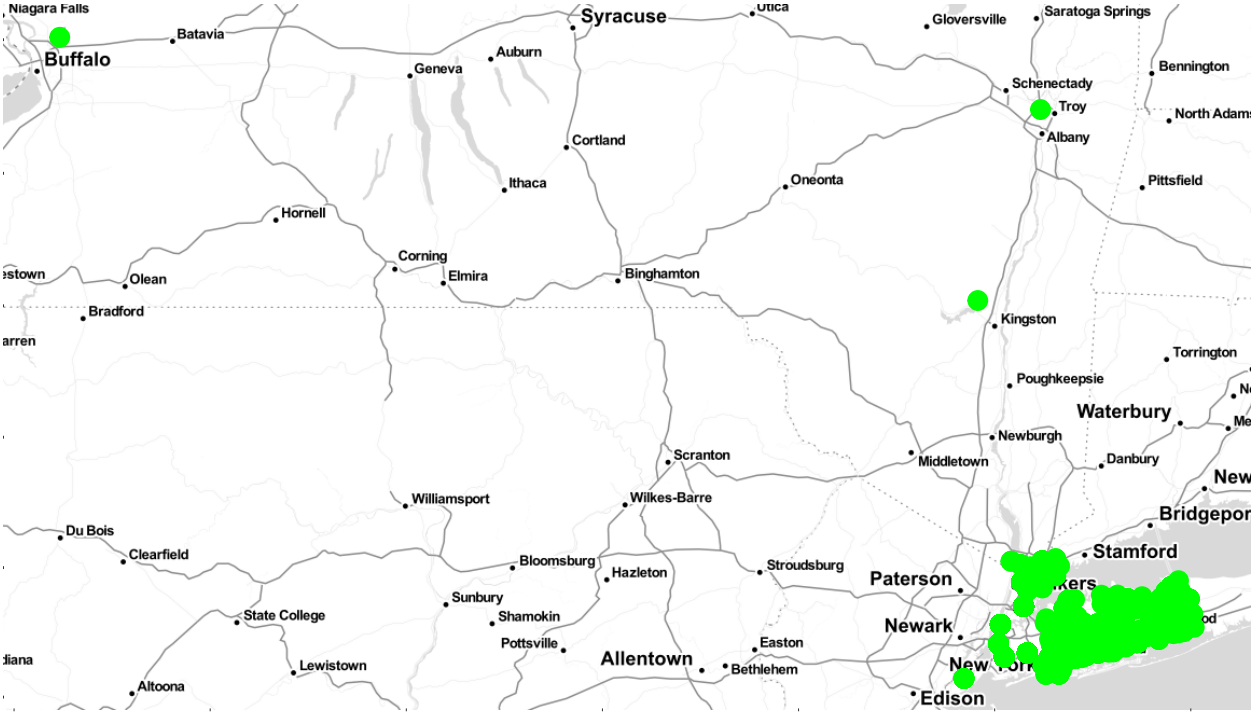
```
qmplot(place_lon, place_lat, data = nj, colour = I('purple'), size = I(3),
       mapcolor = "bw", main="NJ")
```

NJ



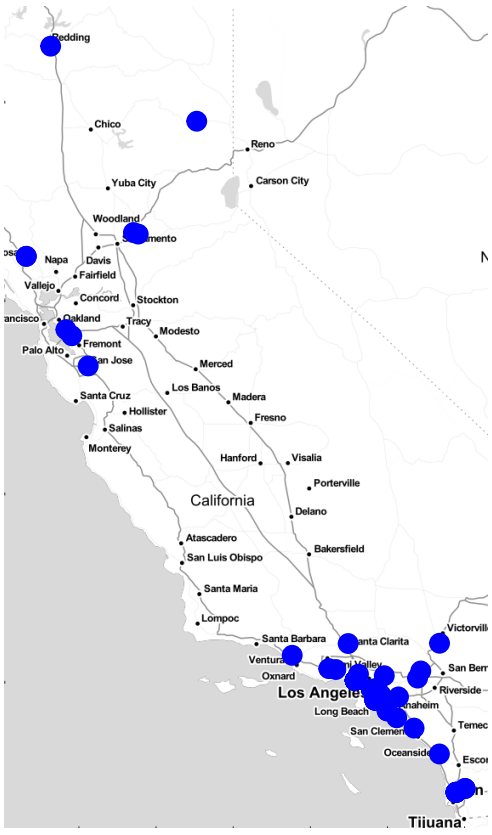

```
qmplot(place_lon, place_lat, data = ny, colour = I('green'), size = I(3),
       mapcolor = "bw", main="NY")
```

NY



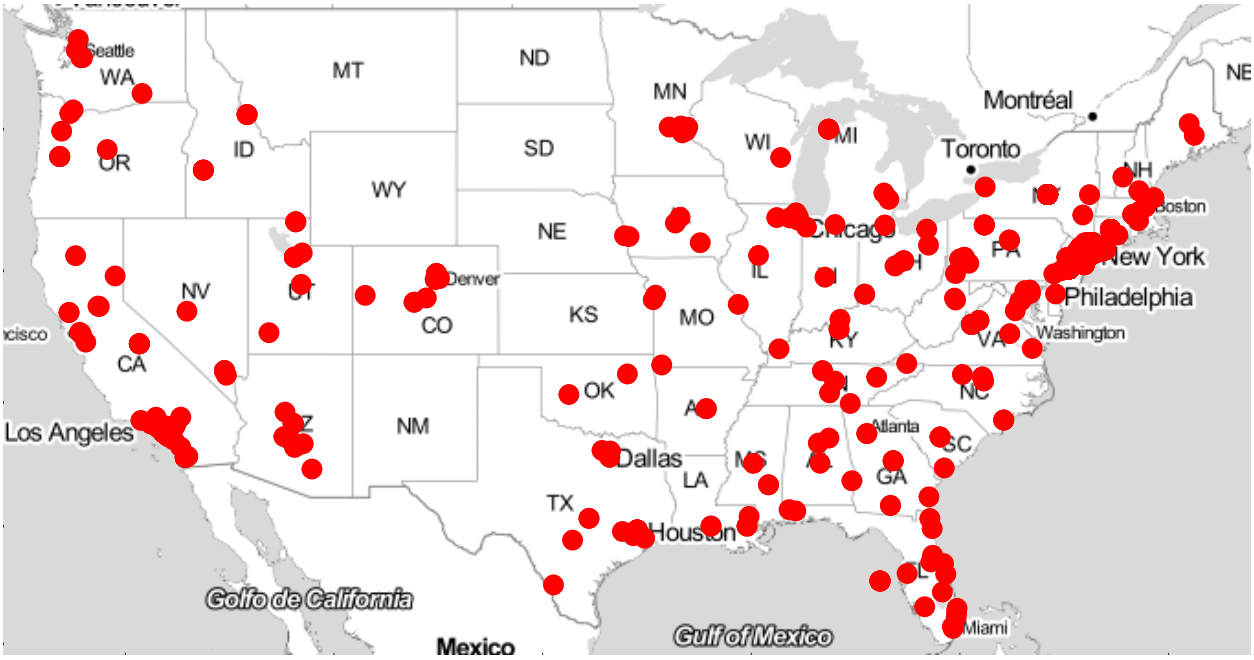

```
qmplot(place_lon, place_lat, data = ca, colour = I('blue'), size = I(3),
       mapcolor = "bw",main="CA")
```

CA



```
qmplot(place_lon, place_lat, data = tweets.df, colour = I('red'), size = I(3),
       mapcolor = "bw",main="USA")
```

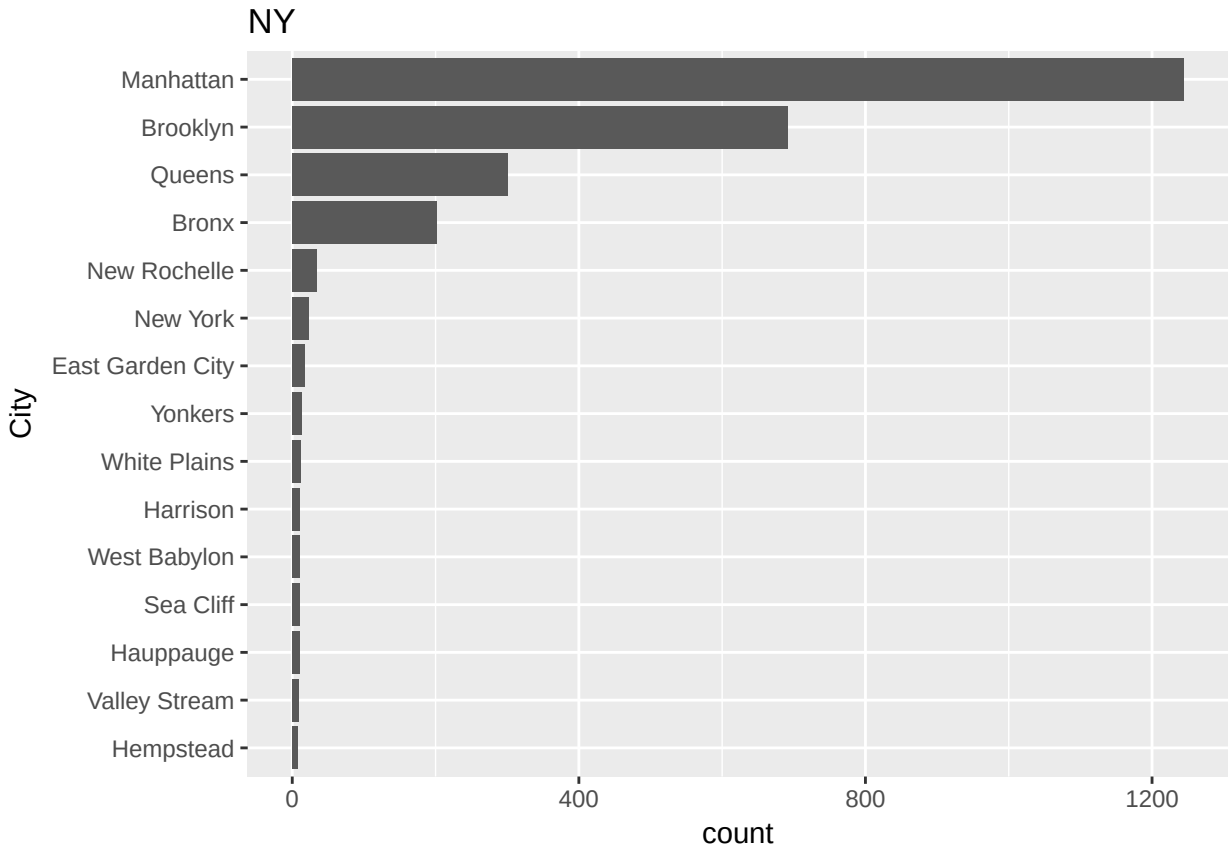
USA



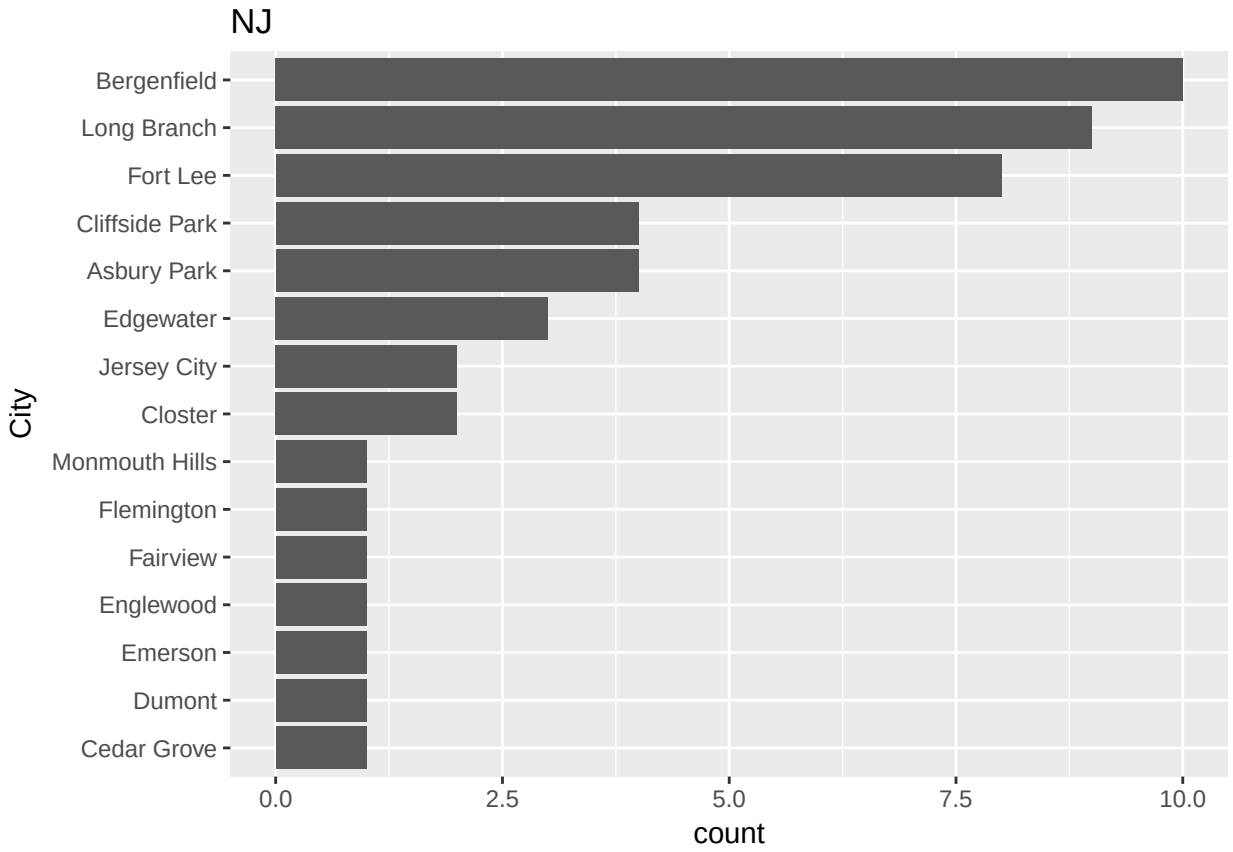

```
#add new variable state
ny[,"state"] <- rep("NY",nrow(ny));
nj[,"state"] <- rep("NJ",nrow(nj));
ca[,"state"] <- rep("CA",nrow(ca));

#get city name
ny$full_name <- as.character(ny$full_name);
nj$full_name <- as.character(nj$full_name);
ca$full_name <- as.character(ca$full_name);
tweets.df$full_name <- as.character(tweets.df$full_name);


get_city <- function(x){
  city_name <- c()
  name <- strsplit(x$full_name,",")
  for(i in 1:nrow(x)){
    city_name <- c(city_name,name[[i]][1])
  }
  return(city_name)
}


# transform to factor
ny[,"city"] <- factor(get_city(ny));
nj[,"city"] <- factor(get_city(nj));
ca[,"city"] <- factor(get_city(ca));
tweets.df[,"city"] <- factor(get_city(tweets.df))
```

```
#combine together
total <- rbind(rbind(ny,nj),ca)
total$lang <- as.character(total$lang)
tweets.df$lang <- as.character(tweets.df$lang)
total_us <- filter(tweets.df,lang=="en")
total <- filter(total,lang=="en")
```

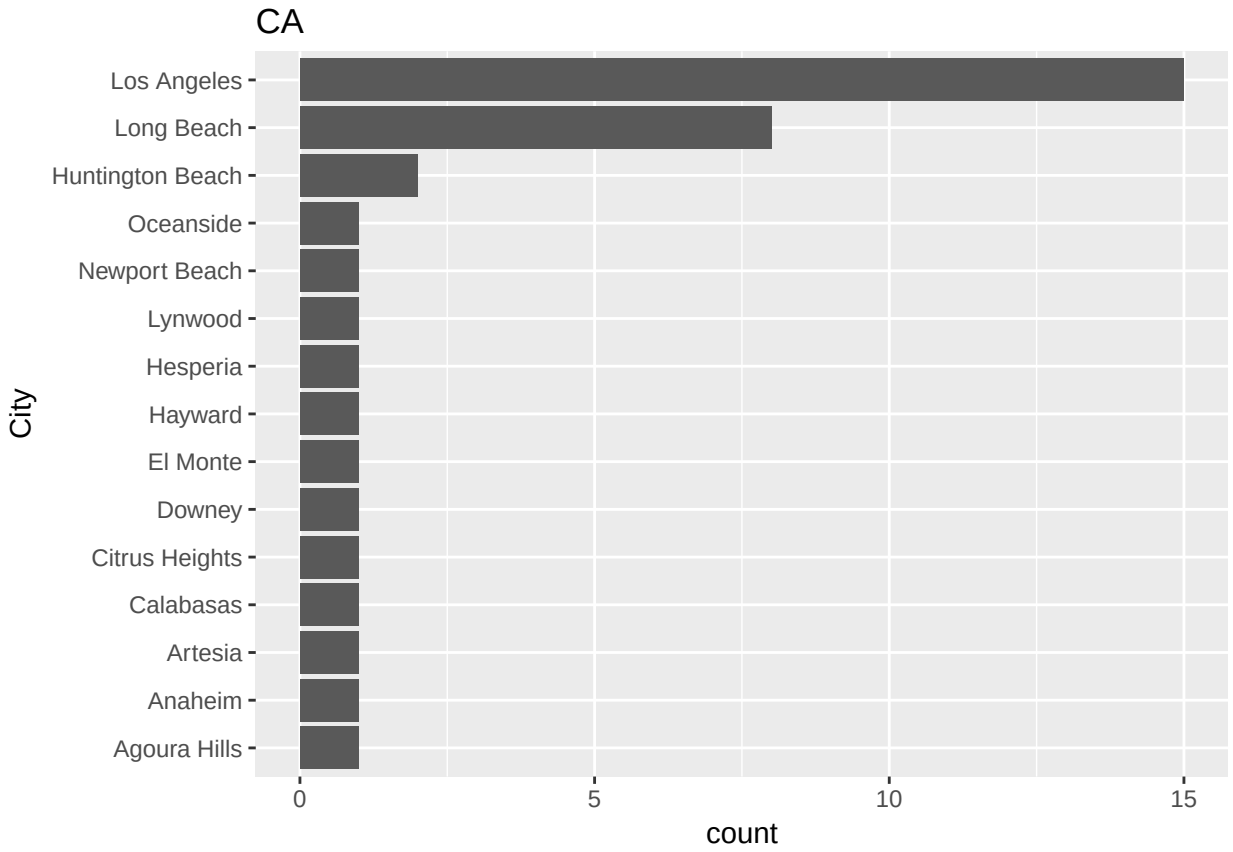
```
# tweet number for every city in state
count <- summary(ny$city)[1:15];ny_city_count <- as.data.frame(count)
count <- summary(nj$city)[1:15];nj_city_count <- as.data.frame(count)
count <- summary(ca$city)[1:15];ca_city_count <- as.data.frame(count)
count <- summary(tweets.df$city)[1:15];us_city_count <- as.data.frame(count)


nycityplot <- ggplot(ny_city_count,aes(reorder(rownames(ny_city_count),count),count))+
  geom_bar(stat = "identity")+coord_flip()+xlab("City")+ggtitle("NY")
nycityplot
```



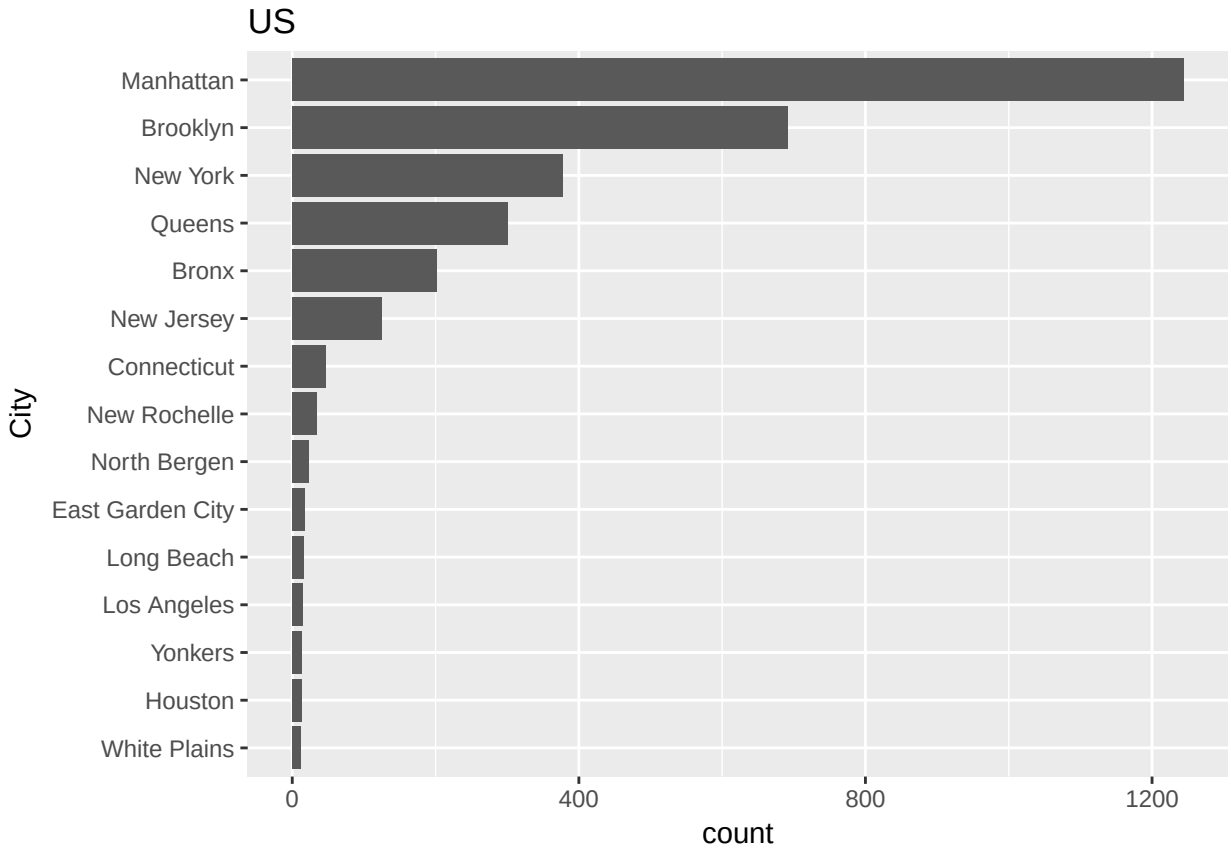

```
njcityplot <- ggplot(nj_city_count,aes(reorder(rownames(nj_city_count),count),count))+
  geom_bar(stat = "identity")+coord_flip()+xlab("City")+ggtitle("NJ")
njcityplot
```



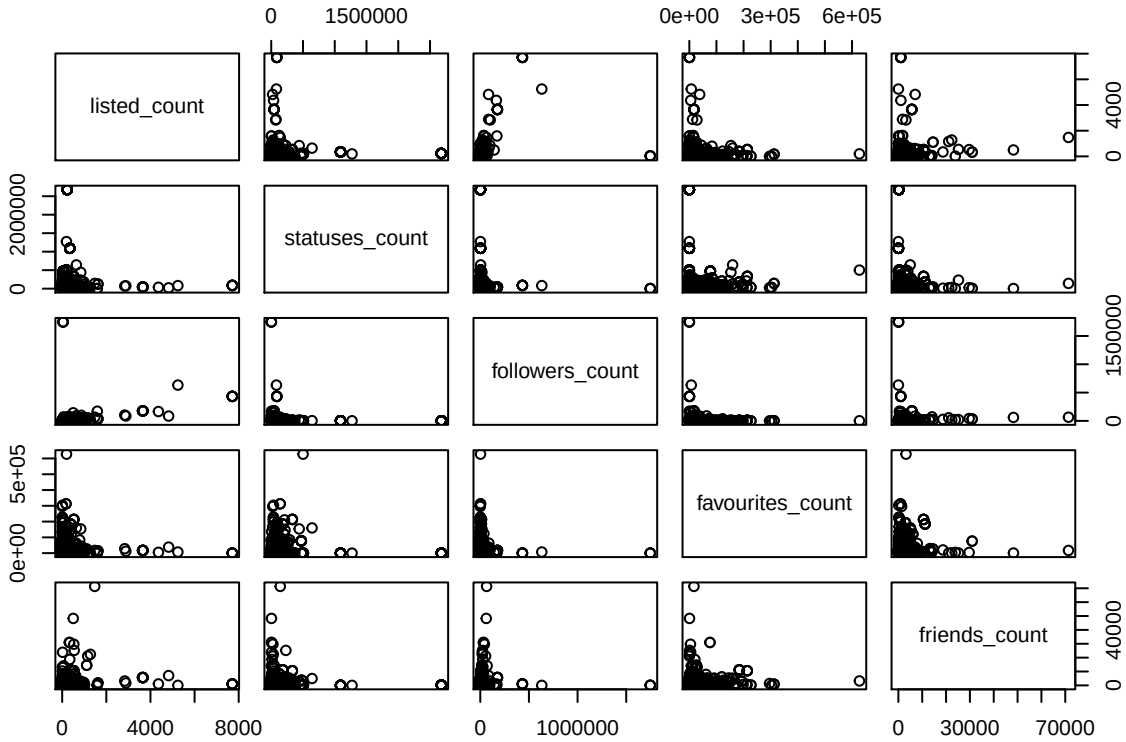

```
cacityplot <- ggplot(ca_city_count,aes(reorder(rownames(ca_city_count),count),count))+
  geom_bar(stat = "identity")+coord_flip()+xlab("City")+ggtitle("CA")
cacityplot
```




```
#grid.arrange(nycityplot, njcityplot, cacityplot, ncol=3)

uscityplot <- ggplot(us_city_count,aes(reorder(rownames(us_city_count),count),count))+
  geom_bar(stat = "identity")+coord_flip()+xlab("City")+ggtitle("US")
uscityplot
```




```
pairs(~listed_count+statuses_count+followers_count+favourites_count+friends_count, data=total_us)
```



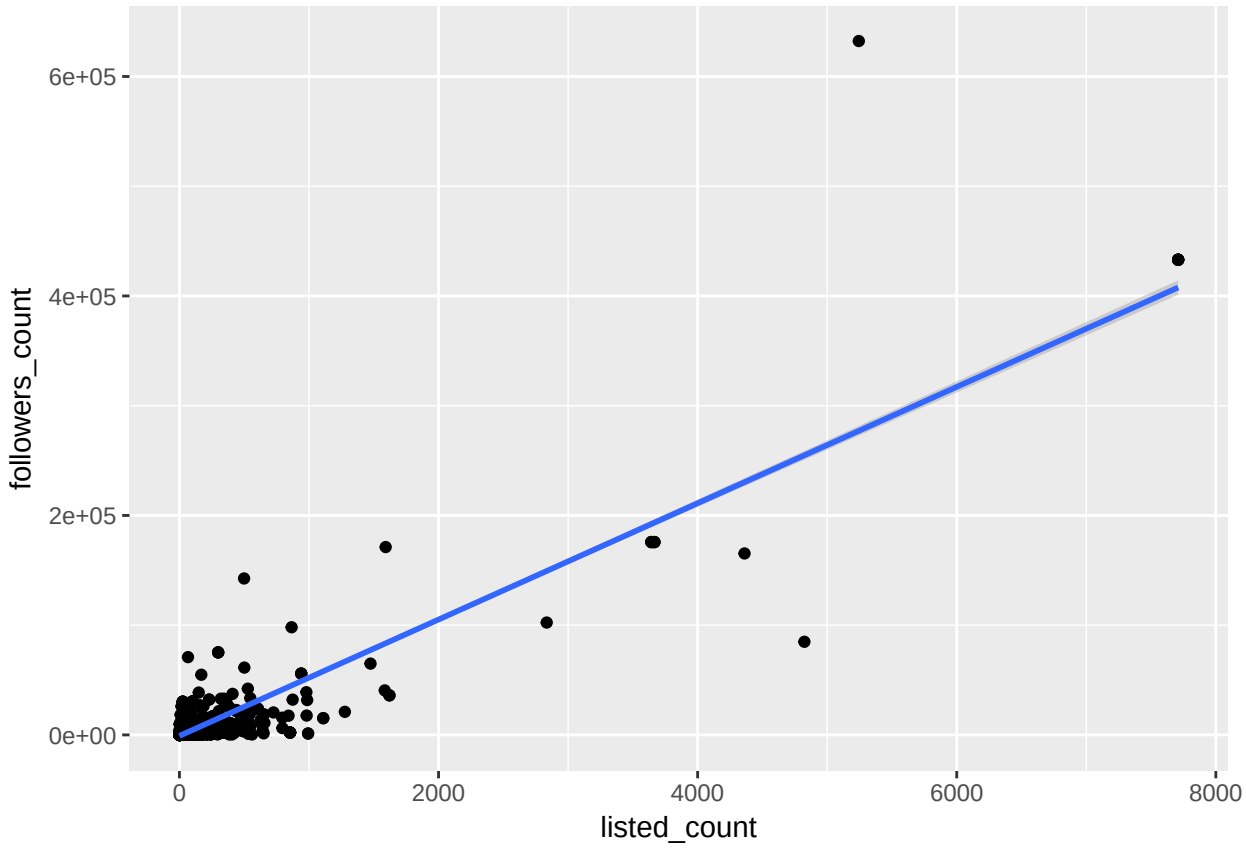
```r
cor(total[,c(3,5,6,7,8)])
```

```
##                  listed_count statuses_count followers_count
## listed_count       1.00000000    0.066180045      0.90859268
## statuses_count     0.06618005    1.000000000      0.01155903
## followers_count    0.90859268    0.011559028      1.00000000
## favourites_count   0.02448727    0.047930213      0.01298768
## friends_count      0.15699332   -0.001003739      0.13210629
##                  favourites_count friends_count
## listed_count           0.02448727   0.156993318
## statuses_count         0.04793021  -0.001003739
## followers_count        0.01298768   0.132106288
## favourites_count       1.00000000   0.151974689
## friends_count          0.15197469   1.000000000
```

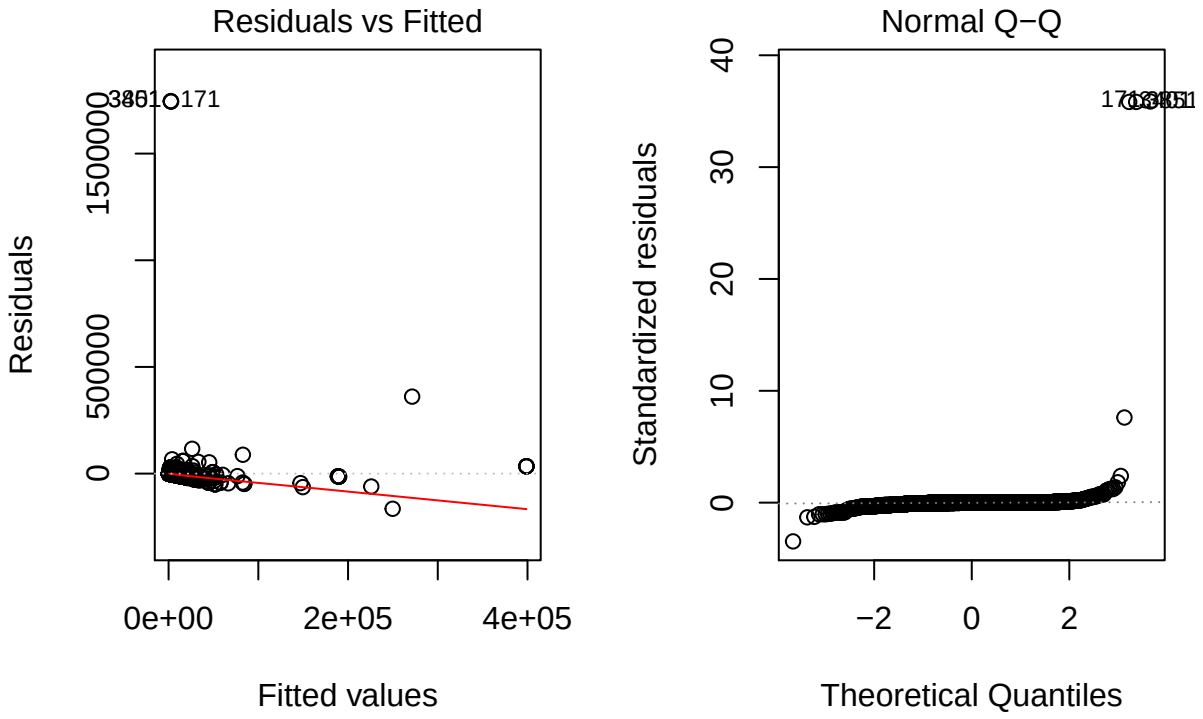
```r
model_fl <- lm(followers_count~listed_count,data=total_us)
summary(model_fl)
```

```
##
## Call:
## lm(formula = followers_count ~ listed_count, data = total_us)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -164852   -1362    -551    -254 1746106
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   532.863    787.149   0.677    0.498
## listed_count   51.660      2.137  24.171   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 48690 on 3991 degrees of freedom
## Multiple R-squared:  0.1277, Adjusted R-squared:  0.1275
## F-statistic: 584.2 on 1 and 3991 DF,  p-value: < 2.2e-16
```

```r
ggplot(total,aes(x=listed_count,y=followers_count))+geom_point()+geom_smooth(method = "lm")
```

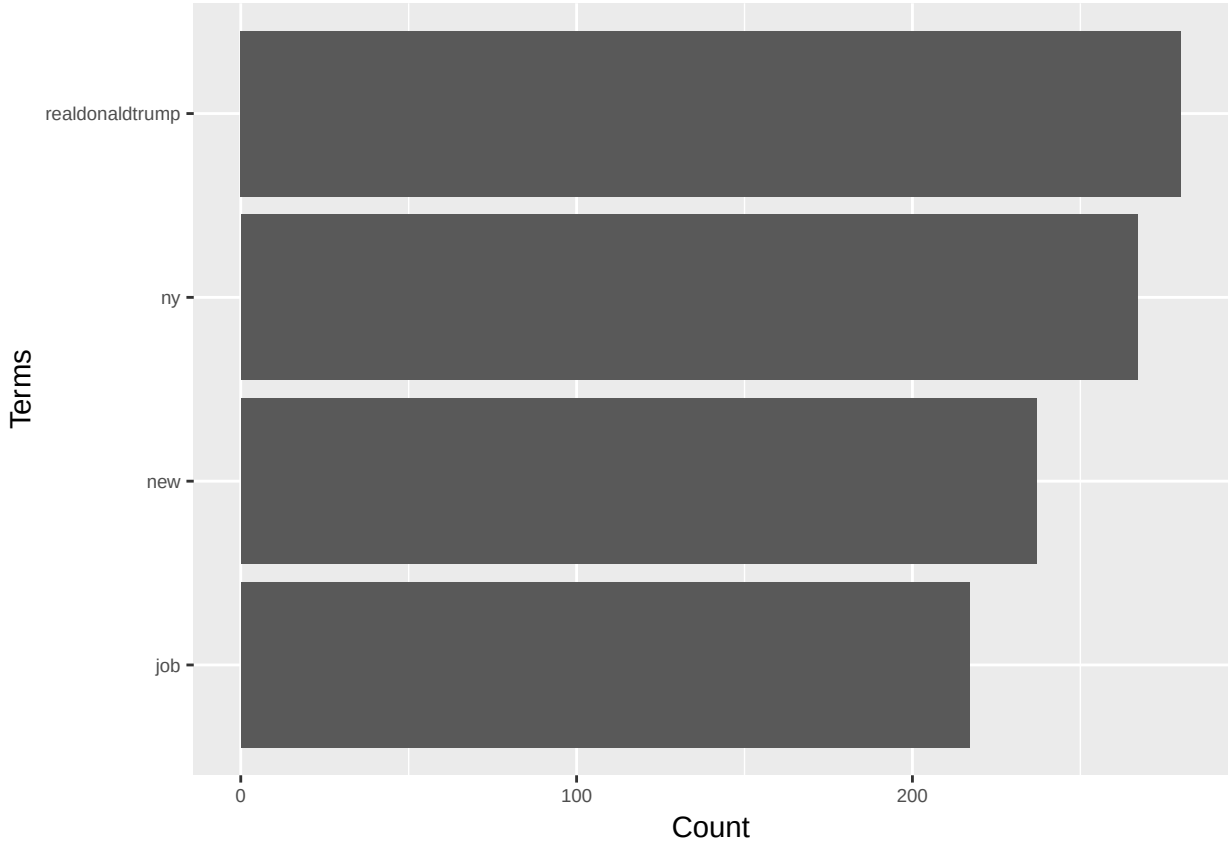



```
par(mfrow=c(1,2))
plot(model_fl,1)
plot(model_fl,2)
```



```r
#wordcloud

tweets_sample.df <- total_us
tweets_sample.df <- tweets_sample.df[tweets_sample.df$lang=="en",]
tweets_sample.df <- tweets_sample.df[tweets_sample.df$geo_enabled==TRUE,]
tweets_sample.df <- tweets_sample.df[tweets_sample.df$place_lat >25 &
                    tweets_sample.df$place_lat <50 & tweets_sample.df$place_lon > -125 &
                    tweets_sample.df$place_lon< -66,]
tweets_sample.df$geo_enabled <- NULL

# build a corpus, and specify the source to be character vectors
myCorpus <- Corpus(VectorSource(tweets_sample.df$text))
# remove anything other than English letters or space(!!!)
removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]*", "", x)
myCorpus <- tm_map(myCorpus, content_transformer(removeNumPunct))
# convert to lower case
myCorpus <- tm_map(myCorpus, content_transformer(tolower))
# remove URLs
removeURL <- function(x) gsub("http[^[:space:]]*", "", x)
myCorpus <- tm_map(myCorpus, content_transformer(removeURL))
# remove stopwords
myStopwords <- c(stopwords('english'),"use", "see", "used", "via", "amp","im")
myCorpus <- tm_map(myCorpus, removeWords, myStopwords)
# remove extra whitespace
myCorpus <- tm_map(myCorpus, stripWhitespace)
# remove punctuation
myCorpus <- tm_map(myCorpus, removePunctuation)
```

```r
# Build Term Document Matrix
tdm <- TermDocumentMatrix(myCorpus, control = list(wordLengths = c(1, Inf)))
term.freq <- rowSums(as.matrix(tdm))
term.freq2 <- subset(term.freq, term.freq >= 200)
df <- data.frame(term = names(term.freq2), freq = term.freq2)
par(mfrow=c(1,1))
ggplot(df, aes(x=term, y=freq)) + geom_bar(stat="identity") + xlab("Terms") +
  ylab("Count") + coord_flip() + theme(axis.text=element_text(size=7))
```




```
m <- as.matrix(tdm)

# calculate the frequency of words and sort it by frequency
word.freq <- sort(rowSums(m), decreasing = T)

# colors
pal <- brewer.pal(8, "Dark2")

# plot word cloud
wordcloud(words = names(word.freq), freq = word.freq, min.freq = 800,
          random.order = F, colors = pal, max.words = 60)
```

```
## Warning in wordcloud(words = names(word.freq), freq = word.freq, min.freq =
## 800, : realdonaldtrump could not be fit on page. It will not be plotted.
```


```
# remove sparse terms
tdm2 <- removeSparseTerms(tdm,sparse = 0.965)
# showing the words that are left gor the analysis
print(dimnames(tdm2)$Terms)
```

```
## [1] "like"            "hiring"          "realdonaldtrump" "just"
## [5] "job"             "ny"              "new"
```

```
m2 <- as.matrix(tdm2)
m3 <- t(m2)   # transpose the matrix to cluster documents
set.seed(122)   # set a fixed random seed
k <- 6   # number of clusters
kmeansResult <- kmeans(m3,k)
round(kmeansResult$centers, digits = 3)   # cluster centers
```

```
##    like hiring realdonaldtrump  just   job    ny   new
## 1 0.005  0.000           0.015 0.036 0.005 0.260 1.184
## 2 0.000  0.003           0.078 0.048 0.000 0.000 0.000
## 3 1.059  0.000           0.086 0.112 0.007 0.000 0.020
## 4 0.016  0.000           0.000 0.016 0.143 1.063 0.016
## 5 0.000  0.438           0.031 0.031 1.016 0.000 0.016
## 6 0.000  1.000           0.000 0.000 0.966 1.021 0.000
```

```
for(i in 1:k){
  cat(paste("cluster ", i, ": ", sep = ""))
  s <- sort(kmeansResult$centers[i,],decreasing = T)
  cat(names(s)[1:5],"\n")
  #print the tweet of every cluster
}
```

```
## cluster 1: new ny just realdonaldtrump like
## cluster 2: realdonaldtrump just hiring like job
## cluster 3: like just realdonaldtrump new job
```

```
## cluster 4: ny job like just new
## cluster 5: job hiring realdonaldtrump just new
## cluster 6: ny hiring job like realdonaldtrump
```

```r
# cluster terms
distMatrix <- dist(scale(m2))
fit <- hclust(distMatrix, method = "complete")
# show cluster dendrogram
p <- plot(fit, xlab="")
p <- rect.hclust(fit, k=6)
```